

Proyecto Final: Sistema de Monitoreo de Distancia con ESP32

Miguel Beltran - 20231005135 & Matteo Rojas - 20231005193 & Juan Vargas - 20232005027

Universidad Distrital Francisco José de Caldas

Ingeniería Electrónica

Palabras clave: ESP32, MicroPython, sensores, servidor web, CSV, IoT.

I. INTRODUCCIÓN

En este proyecto se desarrolla un sistema de monitoreo de distancia basado en una tarjeta ESP32 y un sensor ultrasónico. El sistema permite medir la distancia a un objeto en tiempo real, enviar los datos a un servidor web mediante WiFi y generar automáticamente un archivo CSV con el histórico de mediciones. Adicionalmente, el servidor web despliega una gráfica que muestra la distancia como función del tiempo, actualizándose continuamente.

II. OBJETIVOS

Objetivo general

- Diseñar e implementar un sistema IoT con ESP32 para la medición y visualización remota de distancia.

Objetivos específicos

- Implementar el sensor ultrasónico HC-SR04 en la ESP32 mediante MicroPython.
- Medir la distancia y enviarla a un servidor web mediante solicitudes HTTP.
- Crear un servidor que almacene los datos recibidos en un archivo CSV.
- Desplegar una página web con una gráfica actualizable automáticamente.

III. MATERIALES Y EQUIPOS

- Tarjeta ESP32.
- Sensor ultrasónico HC-SR04.
- Entorno de desarrollo Thonny.
- Computador con conexión WiFi.
- Jumpers.
- Protoboard.
- Servidor local.

IV. MARCO TEÓRICO

A. ESP32

El ESP32 es un microcontrolador que integra capacidades WiFi y Bluetooth, muy utilizado en el desarrollo de aplicaciones IoT debido a su alto rendimiento y bajo consumo energético.

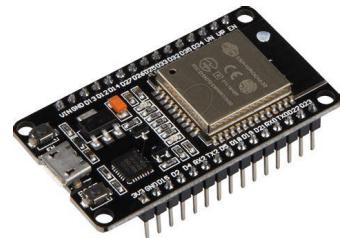


Fig. 1. ESP32

B. MicroPython

MicroPython es una implementación ligera del lenguaje Python diseñada para microcontroladores. Permite programar hardware directamente con sintaxis Python, facilitando el desarrollo rápido.

C. Sensor Ultrasónico HC-SR04

El HC-SR04 permite medir distancias mediante pulsos ultrasónicos. Funciona enviando un pulso por el pin *TRIG* y midiendo el tiempo que tarda el eco en regresar por el pin *ECHO*.



Fig. 2. HC SRO4

D. Servidores Web e IoT

Un nodo IoT puede enviar datos a un servidor mediante protocolos como HTTP. El servidor almacena la información y la presenta mediante una página web.

V. DESARROLLO DE LA PRÁCTICA

Para la lectura del sensor en la ESP32 se utilizó una librería MicroPython que permite medir la distancia en centímetros. La ESP32 se conecta a una red WiFi y envía los datos mediante solicitudes HTTP POST al servidor desarrollado en Python con Flask.

En el servidor, cada lectura recibida se almacena en un archivo `distancias.csv`. La página web asociada carga dicho archivo y genera una gráfica usando la librería `Chart.js`, permitiendo visualizar la distancia en tiempo real.

Código de la ESP32

[https://github.com/SokaLulo/
proyectosProgramacionAplicada/blob/main/
PFAESP32](https://github.com/SokaLulo/proyectosProgramacionAplicada/blob/main/PFAESP32)

Código Servidor

- [https://github.com/SokaLulo/
proyectosProgramacionAplicada/blob/
main/server](https://github.com/SokaLulo/proyectosProgramacionAplicada/blob/main/server)
- [https://github.com/SokaLulo/
proyectosProgramacionAplicada/blob/
main/script](https://github.com/SokaLulo/proyectosProgramacionAplicada/blob/main/script)

Montaje

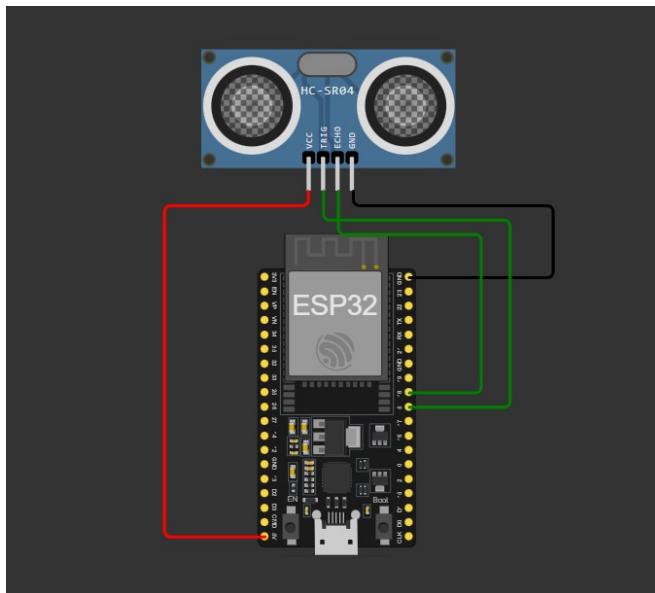


Fig. 3. Equematico

VI. CONCLUSIONES Y RECOMENDACIONES

- El uso de MicroPython facilita la programación de la ESP32 al permitir una sintaxis clara y concisa.
- La arquitectura IoT implementada demuestra cómo un sensor puede integrarse a un sistema web en tiempo real.